1. An XML query rewrite method processing an XML query, the XML query rewrite method comprising:

identifying an expression of the XML query that receives an expression input;

determining whether a set of items defined by the output of the expression is independent of grouping of items in the expression input; and

transforming the expression input into a transformed data stream input, the transforming being conditioned upon the set of items defined by the output of the expression being independent of grouping of items in the expression input.

2. The XML query rewrite method as set forth in claim **1**, further comprising:

further conditioning the transforming on each of a plurality of consumers of the output of the expression receiving a data stream in which each element of the data stream is a single XML item.

3. The XML query rewrite method as set forth in claim **2**, further comprising:

further conditioning the transforming on the expression input not being received by any other expression.

4. The XML query rewrite method as set forth in claim **1**, wherein the identifying of an expression of the XML query that receives an expression input comprises:

identifying an expression of the XML query that receives a sequence of concatenated XML items.

5. The XML query rewrite method as set forth in claim **4**, wherein the XML query is an XQuery, the sequence of concatenated XML items is produced by an XQuery LET clause, and the transforming of the sequence of concatenated XML items into a transformed data stream input comprises:

transforming the sequence of concatenated XML items into a data stream of individual XML items having a format corresponding to an XQuery FOR clause.

6. The XML query rewrite method as set forth in claim **4**, wherein the determining of whether a set of items defined by the output of the expression is independent of grouping of items in the expression input comprises:

determining a granularity property according to:

$$SET(EVAL(\mathrm{exp}, LET(I))) = SET\left(\underset{i \text{ in } \mathrm{LET}(I)}{CONCAT}(EVAL(\mathrm{exp}, i)) \right)$$

where exp represents the expression, I represents the sequence of concatenated XML items, CONCAT is a concatenation operator, EVAL is an expression evaluation operator, SET is a set operator, LET produces a concatenated sequence of XML items, and wherein the set of items defined by the output of the expression is determined to be independent of grouping of items in the expression input if the granularity property is true.

7. The XML query rewrite method as set forth in claim **6**, wherein the transforming of the sequence of concatenated XML items into a transformed data stream input comprises:

transforming the sequence of concatenated XML items into a data stream of individual unconcatenated XML items.

8. The XML query rewrite method as set forth in claim **6**, further comprising:

determining a duplication property according to:

$\forall i, j$ in LET(I) with $POSITION(LET(I), i) \neq POSITION(LET(I), j)$ holds $SET(EVAL(\exp, i)) \cap SET(EVAL(\exp, j)) = \{\}$, and

determining an ordering dependence property according to:

$$\forall i, j \text{ in LET(I) holds } POSITION(LET(I), i) < POSITION(LET(I), j) \Rightarrow \neg \exists e_i \text{ in } EVAL(\exp, i), e_j \text{ in } EVAL(\exp, j) \text{ such that } e_i >> e_j \quad ,$$

wherein POSITION returns ordinal position of an element within a sequence, and the transforming is further conditioned upon the duplication property being true and upon the ordering dependence property being true.

9. The XML query rewrite method as set forth in claim **4**, further comprising:

determining a duplication property that is true if each item contained in the input sequence input to the expression produces unique output items in the output of the expression; and determining an ordering dependence property that is true if an ordering of items in the sequence of concatenated XML items induces a corresponding ordering of items in the output of the expression that adheres to a selected order;

wherein the transforming is further conditioned upon the duplication property being true and upon the ordering dependence property being true.

10. The XML query rewrite method as set forth in claim **4**, further comprising:

determining a duplication property that is true if each item contained in the input sequence input to the expression produces unique output items in the output of the expression;

conditional upon the duplication property being false, inserting an intermediate operation in the XML query that removes duplicate items in the output of the expression; and

performing another query rewrite operation enabled by at least one of the transforming and the inserting.

11. The XML query rewrite method as set forth in claim **4**, further comprising:

determining an ordering dependence property that is true if an ordering of items in the sequence of concatenated XML items induces a corresponding ordering of items in the output of the expression that adheres to a selected order;

conditional upon the ordering dependence property being false, inserting an intermediate operation in the XML query that sorts items in the output of the expression to produce a selected ordering of items in the output of the expression; and

performing another query rewrite operation enabled by at least one of the transforming and the inserting.

12. The XML query rewrite method as set forth in claim **1**, wherein the determining of whether a set of items defined by the output of the expression is independent of a grouping of items in the expression input comprises:

determining a granularity property including at least:

$$SET\left(\underset{p \text{ in } P(I)}{CONCAT}(EVAL(\exp, p)) \right) = SET\left(\underset{p \text{ in } \mathrm{I}}{CONCAT}(EVAL(\exp, p)) \right)$$

where exp represents the expression, I represents the expression input, CONCAT is a concatenation operator, EVAL is an expression evaluation operator, P(I) denotes a partitioning of the expression input I, and SET is a set operator, and wherein the set of items defined by the output of the expression is determined to be independent of grouping of items in the expression input if the granularity property is true.

13. The XML query rewrite method as set forth in claim **12**, wherein the transforming of the expression input into a data stream input comprises:

removing an imposition of a grouping of XML items corresponding to a quantifier that imposes a selected grouping of XML items on the expression input.

14. The XML query rewrite method as set forth in claim **13**, wherein the XML query is an XQuery, and the removing of an imposition of a grouping of XML items corresponding to a quantifier comprises:

removing an imposition of a grouping of XML items corresponding to a quantifier selected from a group consisting of an XQuery LET quantifier and an XQuery FOR quantifier.

15. The XML query rewrite method as set forth in claim **12**, further comprising:

determining a duplication property according to:

$\forall p_i, p_j$ in I holds $SET(EVAL(\exp, p_i)) \cap SET(EVAL(\exp, p_j)) = \{\}$

and determining an ordering property according to:

$\forall p_i, p_j$ in I:

p_i occurs before p_j in I

$\Rightarrow \neg\exists e_i$ in $EVAL(\exp, p_i), e_j$ in $EVAL(\exp, p_j)$ such that $e_i >> e_j$

wherein the transforming is further conditioned upon the duplication property being true and upon the ordering dependence property being true.

16. The XML query rewrite method as set forth in claim **1**, further comprising:

subsequent to the transforming, applying the XML query to a sequence of XML items selected from a group consisting of:

a sequence of XML items extracted from one or more extensible markup language (XML) documents, a sequence of XML items supplied by an Internet service,

a sequence of HTML items extracted from one or more hypertext markup language (HTML) documents, and

a sequence of HTML items supplied by an Internet service.

17. The XML query rewrite method as set forth in claim **1**, further comprising:

further conditioning the transforming on partitioning of the expression output having no effect on any consumer of the expression output.

18. An XML query compilation processor comprising:

an execution compiler transforming an XML query into an executable XML query plan; and

a query rewrite processor performing query transformations on the XML query, said query transformations including transforming an expression input received by an expression that produces an output stream of individual XML items conditional upon a set of items defined by the output stream of individual XML items being independent of grouping of items in the expression input.

19. The XML query compilation processor as set forth in claim **18**, further comprising:

a parser receiving the XML query as a textual XML query and producing a parsed XML query, the query rewrite processor operating on one of the parsed XML query and the executable XML query plan.

20. The XML query compilation processor as set forth in claim **18**, further comprising:

a data flow model processor converting the XML query into a data flow model, the query rewrite processor performing the query transformations on the data flow model, the execution compiler transforming the data flow model into the executable XML query plan subsequent to the query rewrite processor performing the query transformations.

21. The XML query compilation processor as set forth in claim **18**, wherein the query rewrite processor is adapted to perform the expression input transformation by:

determining a granularity property according to:

$$SET(EVAL(\text{exp}, LET(I))) = SET\left(\underset{i \text{ in LET}(I)}{CONCAT}(EVAL(\text{exp}, i)) \right)$$

where I represents the expression input, SET is a set operator, CONCAT is a concatenation operator, EVAL is an expression evaluation operator, LET concatenates XML items of expression input I into a sequence, and exp represents the expression receiving the expression input; and

transforming the expression input into a transformed data stream input, the transforming being conditioned upon the granularity property being true.

22. The XML query compilation processor as set forth in claim **21**, wherein the query rewrite processor is adapted to perform the expression input transformation by further:

determining a duplication property for each consumer of the output stream of individual XML items indicating whether duplicates in the output stream of individual XML items are problematic for that consumer; and

determining an ordering dependence property for each consumer indicating whether a selected ordering is required for that consumer; and

applying one or more corrections including at least one of:

removing duplicates from the output stream of individual XML items flowed to a consumer for which said duplicates are problematic, and

sorting the output stream of individual XML items into a selected ordering required for a consumer.

23. The XML query compilation processor as set forth in claim **21**, wherein the XML query compilation processor processes XQuery XML queries, the expression input is a concatenated sequence input corresponding to an XQuery LET clause, and the query rewrite processor is adapted to perform the expression input transformation by further:

transforming the concatenated sequence input into a data stream of individual unconcatenated XML items corresponding to an XQuery FOR clause.

24. The XML query compilation processor as set forth in claim **18**, wherein the query rewrite processor is adapted to perform the transforming of an expression input by:

determining a granularity property including sub-properties (i), (ii), and (iii) according to:

(i) $SET\left(\underset{p \text{ in } P(I)}{CONCAT}\left(EVAL(\exp, p)\right)\right) = SET\left(\underset{p \text{ in } I}{CONCAT}\left(EVAL(\exp, p)\right)\right)$,

(ii) $\underset{p \text{ in } P(I)}{CONCAT}\left(EVAL(\exp, p)\right)$ is duplicate-free, and

(iii) the XML items in $\underset{p \text{ in } P(I)}{CONCAT}\left(EVAL(\exp, p)\right)$ are in the selected ordering,

where I represents the expression input, P(I) is a partition operator, SET is a set operator, CONCAT is a concatenation operator, EVAL is an expression evaluation operator, exp represents the expression receiving the expression input, sub-property (ii) being included conditional upon a duplicate-free output stream being required, and sub-property (iii) being included conditional upon a selected ordering of the output stream being required; and

removing an imposition of a grouping of XML items on the expression input, the removing being conditioned upon the granularity property meeting a selected criterion.

25. An article of manufacture including a program storage medium encoding instructions executable by an associated computer, said instructions embodying a query rewrite method comprising:

identifying an expression of an XML query having an input data stream grouped by a first quantifier and an output data stream grouped by a FOR quantifier;

computing at least one evaluation property based at least on the expression and the first quantifier; and performing a selected quantifier transformation conditional upon the computed evaluation property.

26. The article of manufacture as set forth in claim **25**, wherein the first quantifier corresponds to a sequence constructor and the performing of a selected quantifier transformation comprises:

replacing the first quantifier by a second quantifier, the second quantifier producing a data stream of individual XML items.

27. The article of manufacture as set forth in claim **25**, wherein the performing of the selected quantifier transformation comprises:

removing the first quantifier.

28. The article of manufacture as set forth in claim **25**, wherein the computing of at least one evaluation property and the performing of a selected quantifier transformation conditional upon the computed evaluation property comprise:

computing properties P1R, P2R, P3R; and

transforming the selected quantifier into a REG quantifier conditional upon the properties P1R, P2R, P3R being satisfied.

29. The article of manufacture as set forth in claim **28**, wherein the performing of a selected quantifier transformation conditional upon the computed evaluation property further comprises:

transforming the selected quantifier into a REG quantifier conditional upon the property P1R being satisfied; and

performing a de-duplication operation conditional upon the property P2R not being satisfied; and

performing a sorting operation conditional upon the property P3R not being satisfied.

30. The article of manufacture as set forth in claim **25**, wherein the computing of at least one evaluation property and the performing of a selected quantifier transformation conditional upon the computed evaluation property further comprise:

computing properties P1F, P2F, P3F; and

transforming the selected quantifier into a FOR quantifier conditional upon the properties P1F, P2F, P3F being satisfied.

31. The article of manufacture as set forth in claim **30**, wherein the performing of a selected quantifier transformation conditional upon the computed evaluation property further comprises:

transforming the selected quantifier into a FOR quantifier conditional upon the property P1F being satisfied; and

performing a de-duplication operation conditional upon the property P2F not being satisfied; and

performing a sorting operation conditional upon the property P3F not being satisfied.